(Preprint) AAS XX-XXX

# GOES-16 ON-STATION CUSTOM MANEUVER GENERATION WITH FOCUSSUITE

**Henry Heim,[*] Natalie Ramos-Pedroza,[*] and Jeffrey Gillette[†]**

This paper discusses the process of the NOAA GOES Mission Operations Support Team (MOST) in mitigating the effects of inconsistently-performing ArcJet Thrusters (AJTs) aboard the GOES-16 spacecraft in geostationary orbit. These AJTs are hot-gas thrusters which augment the gas flow using an electric arc to increase the exit velocity of the gases. Fuel flow problems have led to GOES-16 being unable to safely use its AJTs for maneuvers at times. This primarily affects North/South Station-Keeping (NSSK) maneuvers, which control inclination and use a majority of the station-keeping fuel budget. As a result, GOES-16 has had to occasionally resort to using non-AJT hot-gas thrusters. These thrusters are not placed as close to GOES-16's centerline, have lower thrust, and have lower specific impulse when compared to AJTs. In addition, MOST's ground systems do not officially support using these non-AJT thrusters for NSSK maneuvers. MOST had to quickly develop workaround solutions using the *FocusSuite Autofocus* programming environment to compute the maneuvers. This paper outlines the mixture of commercial off-the-shelf (COTS) and MOST-developed products which have allowed GOES-16 to continue station-keeping even with inconsistently-performing AJTs, showing that script-based automation systems like *Autofocus* can simplify complex solutions to a few scripted user inputs.

## INTRODUCTION

The Geostationary Operational Environmental Satellite-R Series (GOES-R) is the current generation of U.S.-owned and operated geostationary weather satellites that feature Earth and solar weather observation capabilities and data availability. The program is managed collaboratively by the National Oceanic and Atmospheric Administration (NOAA) and the National Aeronautics and Space Administration (NASA) whereby NOAA is the satellite owner and operator, and NASA provides support services. Satellite operations are primarily conducted from the NOAA Satellite Operations Facility in Suitland, Maryland. The GOES satellite fleet has provided Earth imagery and space weather since 1975.

The first satellite of the GOES-R series, also called GOES-R, launched in November 2016. A second, GOES-S, launched in March 2018. GOES-T is planned for launch in March, 2022 and GOES-U is planned for launch in 2024.[1] As these satellites reach their operational orbits, they are renamed to GOES-16, GOES-17, GOES-18, and GOES-19, respectively.

[*] GOES-R Navigation Engineer, Relative Dynamics, Inc., 6401 Golden Triangle Dr. STE 201 Greenbelt MD
[†] Senior GNC Engineer, Relative Dynamics, Inc., 6401 Golden Triangle Dr. STE 201 Greenbelt MD

## Thrusters and Station-Keeping

Due to the tight positioning and vibrational requirements of the science instruments on board GOES-R series spacecraft, as well as their asymmetric solar cell layout, GOES-R series satellites perform roughly ten planned maneuvers each week. Two to three times a week East-West Station-Keeping (EWSK) maneuvers are performed to control the spacecraft's longitude, and North-South Station-Keeping (NSSK) maneuvers are performed at a similar pace to control the spacecraft's inclination. Unlike some geostationary satellites, GOES-R series spacecraft are intended to have no or minimal analemma: in nominal operations, maneuvers are planned such that the inclination is and stays as close to zero as possible.[2]

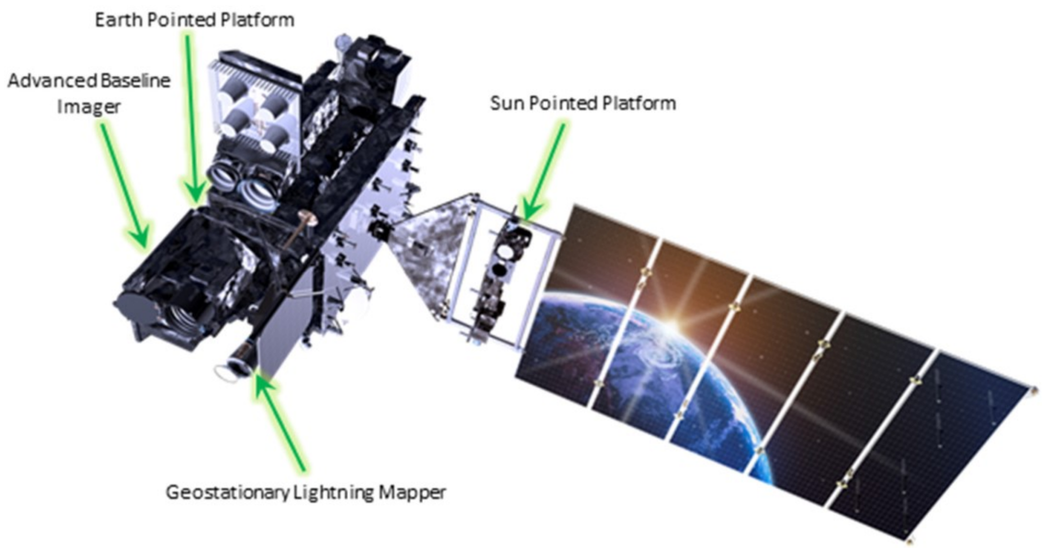


**Figure 1. Visualization of a GOES-R Series Spacecraft in Operational Configuration.[3]**

On average, GOES-R series spacecraft need to perform a Momentum Adjust (MA) maneuver to manage their momentum state every day — this is primarily because of their large and asymmetric solar array, as seen in Figure 1. MA maneuvers are performed in a closed-loop fashion on board the spacecraft; the GOES-R Ground System (GS) sends the spacecraft a desired final momentum state and it is up to the spacecraft to determine which thrusters to use to achieve that state.

In addition, one of the design goals of GOES-R series spacecraft was to allow on-board science instruments to continue collecting data while the satellite performed station-keeping maneuvers.[4] As such, the spacecraft does not slew to perform station-keeping maneuvers and uses relatively low-force thrusters to keep vibrations to a minimum.

**Table 1. Breakdown of AJT and LTR Thruster Count and Intended Use in GOES-R Series Spacecraft.**

| Thruster Type | Count | Intended Use |
|---|---|---|
| Low-Thrust Reaction Engine Assembly (LTR) | 16 | MA maneuvers and EWSK maneuvers |
| ArcJet Thruster (AJT) | 4 | NSSK maneuvers |

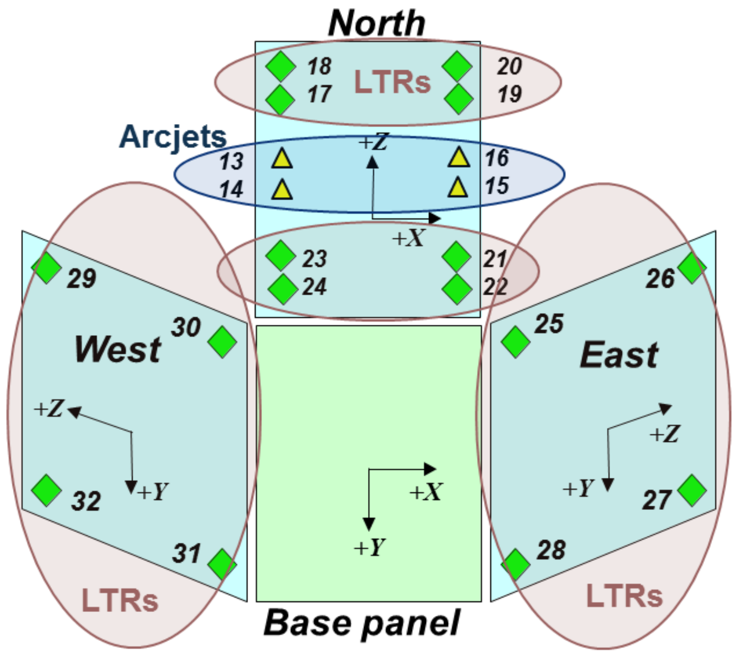


**Figure 2. Locations of AJT and LTR Thrusters for GOES-R Series Spacecraft.**[3]

GOES-R series spacecraft have 20 thrusters intended for use during station-keeping maneuvers, detailed in Table 1. Figure 2 shows the layout of these thrusters on each side of the box-shaped spacecraft. There are no thrusters on the South face, as their exhaust would impinge on the solar array, nor are there any thrusters on the nadir-pointing face which is occupied by imaging instrumentation. There are no station-keeping thrusters on the zenith-pointing face, labeled "Base panel" in Figure 2, as this face contains the thrusters used for relocation from one longitude to another and for disposal to a graveyard orbit at the end of the spacecraft's mission.

AJTs are LTR thrusters which use an electric arc to increase the exit velocity of the propellant. As such, AJTs have higher specific impulse and are more fuel-efficient than LTR thrusters. Because most of the fuel used each year is used during NSSK maneuvers, efficiency in this area is important. The LTR thrusters on the North face of the spacecraft are primarily intended for roll control during MA maneuvers. Additionally, the AJTs are placed closer to the Y-axis centerline of the spacecraft than the North-face LTR thrusters to minimize the torque generated by minute imbalances in AJT alignment. Nominally, the North-face LTR thrusters should only be used when the spacecraft's closed-loop MA routines determine that they are needed to offset momentum in the spacecraft's body Z-axis. Meanwhile, the AJTs are nominally fired for many hours each year to control the spacecraft's inclination, so the AJTs were given priority in placement.

While the AJTs are on the "North" face of the spacecraft, GOES-R series spacecraft have the ability to "yaw flip," inverting the spacecraft so the North face is pointing south, the East face is pointing west, and so on. This can improve thermal control for the spacecraft during some parts of the year. The spacecraft can control its inclination using thrusters pointed in only one direction by maneuvering at either the ascending or descending node, depending on which direction the North face is facing.

**Table 2. Station-Keeping Fuel Usage for GOES-16 and GOES-17 During Their First Full Years On-Orbit.**

| Spacecraft | MA Fuel Usage (kg) | EWSK Fuel Usage (kg) | NSSK Fuel Usage (kg) | Total Station-Keeping Fuel Usage (kg) |
|---|---|---|---|---|
| GOES-16 | 7.49 | 6.20 | 31.55 | 45.24 |
| GOES-17 | 5.92 | 5.99 | 33.23 | 45.14 |

Table 2 shows station-keeping fuel usage for GOES-16 and GOES-17 during 2017 and 2019, respectively, which were each spacecraft's first full year on-orbit. The table compares each spacecraft's first full year on-orbit because during those years the spacecraft had comparable fuel loads. The table illustrates that most of the fuel used each year is for NSSK inclination control. The values differ between the spacecraft because of the different operational longitudes, the different relative positioning of the moon between the two years, and the slightly different checkout procedures for the two spacecraft.

### Maneuver Planning

The GOES-R GS primarily relies on a customized version of the COTS software product *FocusSuite* by GMV to plan maneuvers. This software suite has many options, can plan maneuvers in a variety of modes, is meant to be used alongside a detailed checklist giving the user exact steps to take to plan maneuvers that are correct, efficient, in the right order, and at the right times. However, *FocusSuite* also has a built-in automation toolbox: *Autofocus*. *Autofocus* has the ability to do basic mathematical operations, call functions, configure *FocusSuite* settings, and execute *FocusSuite* modules.

The MOST Navigation Team heavily uses *Autofocus* to support the NOAA maneuver planners. *Autofocus* scripts are scheduled to start themselves so that, if all goes well, the scripts will be complete by the time the planners arrive in the morning to manually check over the already-generated plots and maneuver plans to verify that the maneuvers which were planned are correct. The MOST Navigation Team has written over 42,000 lines of *Autofocus* code to support this process, and in nominal operations the system works well enough to support the mission.

## CUSTOM MANEUVER GENERATION

Unfortunately, operations are not always nominal. The version of *FocusSuite* used by the GOES-R program has been customized by GMV and L3Harris (the GOES-R Ground System primary contractor) to have GOES-R-specific functionality. For example, when using the *MANCMD - Thruster Activity Prediction and Maneuver Commands* module and selecting thruster sets from drop-down menus, the menus are populated with the thruster sets that GOES-R satellites use. Each thruster has corresponding settings, pages, and tabs in the *Propulsion Database* which define thruster properties, and *FocusSuite* has internal checks to ensure that each maneuver is only using valid thruster sets for that maneuver type (NSSK, EWSK, or MA).

Those checks can prevent users from making basic errors like using an East- or West-face thruster set to perform an NSSK maneuver, but they restrict users' options if a spacecraft needs to use a thruster set for something other than the thruster set's intended purpose. A Work Request

(WR) process would be required, followed by extensive tests before the updated version of *FocusSuite* would be accepted on the GOES-R Ground System. Operational concerns regarding station-keeping may require a faster response for some contingencies than a formal WR process can provide.

**Contingency Thruster Selection**

Erratic performance of both of GOES-16's AJT pairs led to a long period in 2020 where the AJTs were being run on lower-power (and, thus, lower-Isp) modes to prevent the hardware from being damaged. The GOES-R MOST was in frequent communication with the spacecraft's integrator, Lockheed Martin, who provided guidance on which AJT pair to run, what power levels to run it at, and what test burns should be performed. However, eventually even this contingency stopped working as intended and the spacecraft began to experience unexpected momentum effects while firing either AJT pair. While the lower Isp was problematic in terms of mission lifetime, the more pressing concern was that GOES-16 and its science instruments were designed to operate at inclinations below 0.1°.[5] If GOES-16's AJT performance continued to degrade, the spacecraft might not have been able to stay inside the operational bounds of some of the science instruments using nominal inclination-control methods.

The MOST Navigation Team began investigating the ability to use nonstandard thruster sets for station-keeping maneuvers in *FocusSuite* once the AJT issues started to surface. After some experimentation inside the *Autofocus* environment, the Navigation Team determined that it would be possible to generate such maneuvers manually and even to automate their creation. That determination, combined with a then-formulating plan to shift away from *FocusSuite* for maneuver planning to a custom system build around NASA's *General Mission Analysis Tool*, led GOES-R MOST management to accept the MOST Navigation Team's *Autofocus* solution to use nonstandard thruster sets for NSSK maneuvers instead of a WR-based process where *FocusSuite* would be updated to allow nonstandard thruster sets to be used.

After conferring with Lockheed Martin, a plan was formed to use the North face diagonally paired LTR thrusters (17/21, 18/22, 19/23, and 20/24; see Figure 2) to replace the AJTs for inclination control maneuvers. Because these thrusters are farther from the center of the North face of the spacecraft, these "LTR NSSK" maneuvers were expected to generate a greater momentum effect than standard ("AJT NSSK") maneuvers. Additionally, since the AJTs are essentially LTR thrusters with a greater exit velocity, the LTR thrusters naturally have lesser thrust. Thus, LTR NSSK maneuvers would either need to have longer durations or occur more frequently than AJT NSSK maneuvers.


**Figure 3. NSSK Maneuver Order.**

One final consideration for creating LTR NSSK maneuvers was that the thrusters used for the NSSK maneuver itself should not be used for the bracketing Pre-MA or Post-MA maneuvers that go along with the NSSK maneuver. Because NSSK maneuvers typically impart a much larger amount of momentum on the spacecraft than EWSK maneuvers or standalone MA maneuvers, MA maneuvers are placed both before ("Pre-MA") and after ("Post-MA") the NSSK maneuver. Figure 3 is a visual representation of how NSSK maneuvers are implemented. The Pre-MA ensures that the NSSK does not overload the spacecraft's RWAs, and the Post-MA brings the spacecraft back to a neutral momentum state from which it can safely endure solar radiation pressure until the next day's MA.

The thrusters used for the NSSK maneuver itself should not be used for the Pre-MA or Post-MA maneuvers to avoid having any thrusters accumulate too much on-time. These thrusters have expected lifetimes measured in total on-time, so overusing them might cause wear-and-tear related problems sooner than would be seen if thrusters are alternated between the bracketing MAs and the LTR NSSK itself.

**Limitations of *Sol***

While *Autofocus* has been an incredibly powerful tool for the MOST Navigation Team in helping to automate use of *FocusSuite*, its domain-specific mini-language *Sol* has a few limitations. The two most important in creating LTR NSSK maneuvers are that there are very few string manipulation functions and that there is no direct support for list indexing.

The former problem mostly presents itself when text needs to be output in a particular format. The only string manipulation function available is concatenation, and as such it can be difficult to properly format numbers in scientific notation or with varying amounts of decimal places. There is a built-in function to round decimal numbers, but it does not always perform as expected. While it took some time to develop effective workarounds — such as using modular arithmetic, multiplication, and conditional statements to try and count decimal places — the issue of being unable to easily manipulate strings has not been overly problematic.

The latter problem, however, presents quite a challenge. While the *Autofocus* documentation does show a way to iterate through a list item-by-item to compute the list's value at a given index, by way of an example script doing just that (see Appendix: Sample *Sol* Script), this method can be extremely slow if a list is large. In most programming languages, lists can be accessed in constant time because the size of each element in the list is known; because each list item in *Sol* can be any size, traditional approaches do not work and the item-by-item scanning method must be used. For large lists, such as the built-in maneuver database, it can take more than a minute for *Sol* to return the last item in the list. This slows down the execution of scripts, as it can sometimes be necessary to access multiple items at the end of a large list.

**Code Snippet 1. Defining a String in *Sol*.**

```
set exampleString to "this is an example string"
```

Code Snippet 1 shows the *Sol* code to define a string. When using the example script showing how *Sol* can iterate through a list (see Appendix: Sample *Sol* Script), the value taken from the list/string in the first iteration will be "this," the value in the second iteration will be "is," followed by "an," "example," and "string." To retrieve a specific index from a list, say the third element ("an"), the script needs to keep track of how many loop iterations have been performed and break out of the loop at the appropriate time. In this example, on the third time through the loop the program will see that the target index has been reached and the correct value "an" will be returned. While this is easy to work around in itself – and opens the possibility for advanced maneuver-planning scripts that would be difficult without the ability to use lists – it adds complexity and length to any script that needs to access a list, which is a very common operation.

A further feature of the string-based lists is that *Sol* natively supports nesting, though this feature is not mentioned in any of its documentation as provided to the MOST Navigation Team and this was discovered via experimentation. If items in a list are bounded by curly brackets, then those items will be considered a single element of the list.

**Code Snippet 2. Defining a Nested String in *Sol*.**

```
set exampleString to "this {is an example} string"
```

Code Snippet 2 shows the *Sol* code to define a nested string. When using the same list-indexing script as before, the first item in the list is “this,” the second item is “is an example,” and the third item is “string.” The addition of the braces changed the string from representing a five-element list to representing a three-element one, and the second item in the list is itself a list that can be separately iterated through; the nested list would have the three elements “is,” “an,” and “example.”

This quirk is how *FocusSuite* represents many of its internal databases in *Autofocus*. For example, the Impulsive Maneuver File (IMF) is returned as a "list of lists", where each item in the top-level IMF list is itself a list containing the details of each maneuver. So, for example, a maneuver-planning script might need to know whether the last maneuver in the IMF has been reconstructed using thruster telemetry data. The script would iterate through the IMF, saving the maneuver on each iteration. Each iteration would overwrite an older maneuver with a newer one, so when *Sol* exits the IMF-iterating loop, the last maneuver in the IMF would be saved. Then that maneuver would be iterated through until the desired index was reached, at which point the desired value would be returned. The fact that the IMF, which is the ultimate source of truth for maneuvers in *FocusSuite*, is simply a string/list is what makes automated LTR NSSK maneuver planning possible.

**Automated LTR NSSK Maneuvers**

As previously mentioned, *FocusSuite* has a hard-coded list of acceptable thruster sets for each maneuver type. NSSK maneuvers must be performed using one of the two AJT thruster sets, otherwise the *MANCMD* module – which translates impulsive maneuvers to “implemented” maneuvers with predicted durations and momentum effects – will fail to run. Thus, the core problem at hand when planning LTR NSSK maneuvers is how to implement a maneuver that *MANCMD* refuses to work with.

The precise steps by which LTR NSSK maneuvers are planned are convoluted and dependent on *FocusSuite*’s specific design choices. First, an impulsive NSSK maneuver is planned just as one would be for a standard AJT NSSK using *FocusSuite*’s *INCLON – Inclination Maneuvers* module. This provides the timing and ΔV requirement of the LTR NSSK maneuver, as impulsive maneuvers are impartial to thruster selection. Next, the maneuver string is modified to trick *FocusSuite* into thinking the maneuver is an EWSK maneuver. The *MANCMD* module is used to implement the maneuver as an EWSK maneuver using East- or West-face LTR thrusters. This provides the duration of the maneuver assuming two LTR thrusters are being used. Because EWSK maneuvers always use two LTR thrusters, and because the LTR thrusters on the North face are identical to those on the East and West faces, this provides an acceptable approximation of North-face LTR thruster performance. The maneuver string is then modified again to convert the entry back into an NSSK maneuver.

At this point, the maneuver is almost complete. The efficient impulsive maneuver time has been found and *FocusSuite*’s thrust models have been leveraged to compute the maneuver duration, from which the maneuver start and end times are easily found.

The only remaining task is to properly compute the maneuver’s momentum effects, as those provided by *MANCMD* are incorrect since they were computed for a pair of East- or West-face LTR thrusters. Computing that momentum only requires a few simple steps. First, knowledge of the spacecraft’s mass, the maneuver’s duration and ΔV, and the number of activated thrusters is used to estimate the force output of each thruster. *FocusSuite* databases provide the precise locations of thrusters and of the spacecraft’s center of mass, and those are used to compute the torque

generated by each thruster. That torque is then multiplied by the maneuver's duration, giving the momentum effect of the maneuver. Finally, the maneuver string is updated with these newly-computed momentums then saved back to the IMF.

These steps produce a maneuver that looks exactly like one would expect an LTR NSSK maneuver to look. The maneuver has predictions for duration, ΔV, and momentum effect in the ECI frame: everything that the rest of the GOES-R GS expects from *FocusSuite*-created maneuvers.

### MA Considerations

Beyond this simplified synopsis, the bracketing Pre-MA and Post-MA maneuvers that help handle the spacecraft's momentum state also need to be created. These maneuvers typically need to use North-face LTR thrusters to reach their desired momentum states, and as such they also contribute North/South ΔV and change the spacecraft's inclination. Because the ΔV value computed by *INCLON* is representative of the total change in velocity needed to achieve the desired orbit, the contribution from the Pre-MA and Post-MA maneuvers needs to be considered in addition to that of the NSSK maneuver itself.

This is done by running the function containing the steps to create the LTR NSSK maneuver in a loop and using a numerical solver (discussed in the next section) to find an NSSK ΔV that results in the total ΔV of the LTR NSSK and its bracketing MAs being acceptably close to the impulsive ΔV computed by *INCLON*. Once the NSSK maneuver itself has been planned, its predicted momentum effect is handed to a function which plans an MA before the LTR NSSK maneuver, then an MA is planned for immediately after the LTR NSSK maneuver. Those maneuvers' ΔV values are retrieved from the IMF, summed together, and compared against the target ΔV computed by *INCLON*.

In addition, the spacecraft's RWAs have limits to how much momentum they can absorb before they become saturated and need to be spun down. The spin-down is performed by the Post-MA maneuver immediately following the LTR NSSK maneuver, but these momentum limits place a practical duration limit (and, thus, a ΔV limit) on each LTR NSSK maneuver. On each loop iteration the total momentum effect of the LTR NSSK maneuver needs to be checked, and if the ΔV computed by *INCLON* requires a maneuver with too high of a momentum effect then a maneuver producing a momentum effect slightly below the maximum will be used; this is the most ΔV the spacecraft can generate with an LTR NSSK without violating flight rules. A reduced-length NSSK maneuver with some inclination effect is better for station-keeping than no NSSK maneuver at all.

Finally, each loop iteration checks that the LTR NSSK's ΔV does not drop below a configurable minimum value. If it does, then an MA maneuver is planned instead; these use LTR thrusters by default and are usually targeted to occur at the same time as NSSK maneuvers, so they also contribute in small ways to inclination control if they happen to use a North-face LTR.

These two *Autofocus* scripts, one which plans the LTR NSSK maneuver itself and one which coordinates the constraints the maneuver needs to be planned under, including planning the Pre-MA and Post-MA, work in tandem to compute maneuvers that are accurate and have the best chance at matching the desired on-orbit performance. The codebase to achieve this is thousands of lines long spread across dozens of functions and files; while the process can be described in a few short paragraphs, the low-level implementation is nontrivial.

### Proportional Variable-Gain Numerical Solver

A two-step process of convergence is used to ensure the maneuvers generated meet ΔV and momentum requirements. Firstly, due to the same RWA limitations that bound the amount of ΔV a LTR NSSK maneuver can achieve, LTR NSSK maneuvers must be regenerated until they are

below the flight limits for momentum. This is done using a proportional variable gain to numerically minimize the error. This proportional momentum gain is reduced whenever an iteration's error in momentum magnitude is greater than that of the last iteration. In other words, if the iteration's momentum exceeded the target value and overshot, the gain is reduced so the next iteration does not exceed the target value in the opposite direction. This prevents a state where the solver is endlessly oscillating over the target value without ever getting closer, while preserving convergence speed by using a large initial gain value.

Once the first solver has satisfied its momentum magnitude limits, an additional solver must converge the overall ΔV (Pre-MA, LTR NSSK, and Post-MA) to match what *INCLON* has computed as the appropriate ΔV for the maneuver. This step can take many iterations, as the Pre-MA and Post-MA maneuvers often contribute significant amounts of ΔV in the North or South directions. An iteration for this solver involves repeating the entire process of the first solver. This ΔV solver uses a similar variable-gain method as the momentum solver, with its own proportional ΔV gain.

When used in tandem, the momentum and ΔV solvers ensure that the generated maneuvers provide the necessary ΔV without overloading the spacecraft's momentum management abilities. The momentum solver is necessary because of the unusual momentum buildup experienced when firing North-face LTR thrusters for up to an hour at a time. The ΔV solver solves a problem that *FocusSuite* normally handles internally, but because the *MANCMD* module is used to generate an interim EWSK maneuver it does not include the Pre-MA and Post-MA as *FocusSuite* does not know that those will eventually be added.

**Effects On Maneuver Cadence**

The spacecraft's RWAs can only absorb so much momentum before they need to be spun down. Because the North-face LTR thrusters are farther from the center of the spacecraft's North face, they cause more of a momentum effect than the more-centered AJTs for an equivalent amount of ΔV because of their longer moment arm. The difference is great enough that when GOES-16 is using LTR NSSK maneuvers, inclination-control maneuvers need to be performed as much as three times as often as when using AJT NSSK maneuvers.

This has resulted in GOES-16 performing far more NSSK maneuvers than it would otherwise. Before any anomalies occurred, GOES-16 nominally had a four-day maneuver cadence: NSSK, MA, EWSK, MA. This was enough to keep the spacecraft in its proper longitude and with very low inclination. When using LTR NSSK maneuvers, however, GOES-16 has typically had a seven-day maneuver cadence: NSSK, NSSK, NSSK, NSSK, NSSK, EWSK, MA. The length, and thus ΔV effect, of individual LTR NSSK maneuvers is limited by how much momentum the spacecraft can handle before it loses attitude control. Since this would lead to the spacecraft entering safe-hold mode, the LTR NSSK maneuver durations are limited during maneuver planning and they need to be performed far more often than AJT NSSK maneuvers in order to keep inclination minimized.

**Results**

GOES-16 has performed 52 LTR NSSK maneuvers to date. After two initial test maneuvers, the remaining 50 LTR NSSK maneuvers were performed during a 90-day timespan ending in August, 2021. During this period, LTR NSSK maneuvers contributed 5.9 m/s of ΔV towards inclination control and the preferred LTR19/LTR22 North-face pair accumulated over 35 hours of on-time each; other testing campaigns were ongoing which occasionally interrupted the expected cadence and which contributed towards inclination control.

LTR NSSK maneuvers largely performed as expected. Two maneuvers were aborted before they could finish due to unacceptable momentum buildup which was incorrectly predicted in *Autofocus* due to the long moment arm of the North-face LTR thrusters relative to that of the AJTs. Once GOES-16 performed a handful of LTR NSSK maneuvers, an *Autofocus* script was made to create new maneuvers which assume identical performance as past maneuvers. This script used the same string-manipulation techniques developed for LTR NSSK maneuver generation, and this method improved the prediction accuracy of LTR NSSK maneuvers; instead of trying to compute what ΔV and momentum effect should be based on evidently-inaccurate databases, real-world data was used instead. This method also greatly reduced how long it takes *Autofocus* to generate an LTR NSSK maneuver. Generating an LTR NSSK maneuver from scratch takes around 40 minutes of computation but creating a new LTR NSSK based on the results of a past one takes less than two minutes.

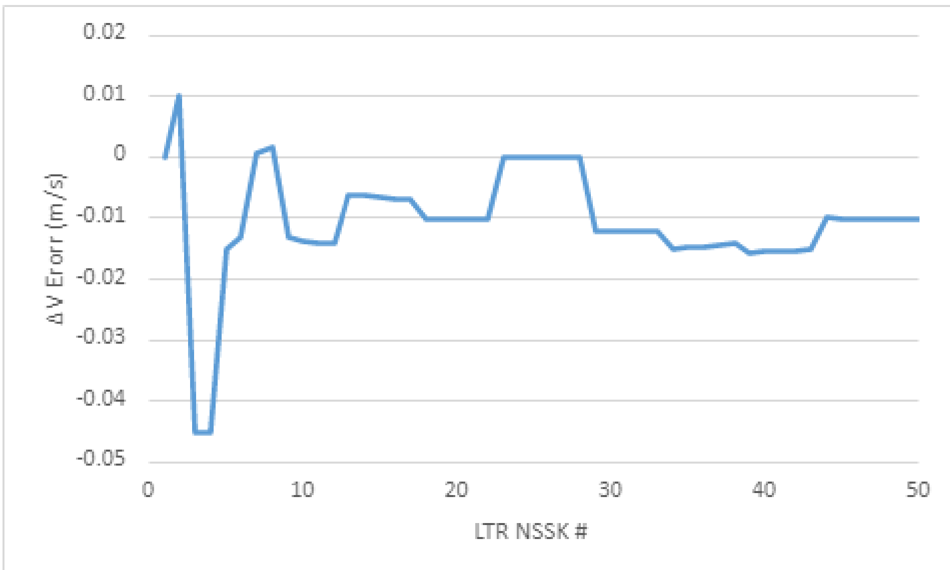


**Figure 4. Difference Between *FocusSuite* ΔV Prediction and Observed Orbit Change for LTR NSSK Maneuvers.**

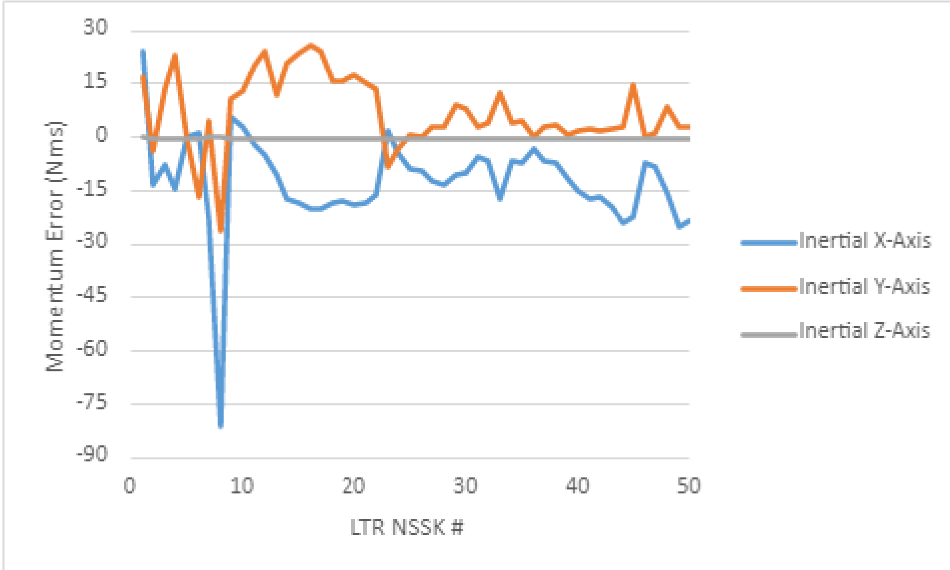


**Figure 5. Difference Between *FocusSuite* Momentum Predictions and Observed Momentum State Changes For LTR NSSK Maneuvers.**

Figure 4 and Figure 5 show that once this new maneuver generation method was adopted, GOES-16 stopped seeing large error outliers when comparing the predicted LTR NSSK performance against reality.

**Table 3. Comparison of 50 LTR NSSK Maneuvers to 50 Nominal AJT NSSK Maneuvers.**

| Thruster Type | Total Duration (h) | Total ΔV (m/s) | Acceleration (m/s$^2$) | Total Fuel Usage | Fuel Efficiency (m/s/kg) |
|---|---|---|---|---|---|
| LTR | 35.91 | 5.904 | $4.567 \times 10^{-5}$ | 12.95 | 0.4559 |
| AJT | 49.21 | 24.42 | $1.378 \times 10^{-4}$ | 15.51 | 1.574 |

Table 3 compares the 50 non-test LTR NSSK maneuvers with 50 nominal AJT NSSK maneuvers. Acceleration is listed to illustrate that the AJTs are much more effective at changing the spacecraft's orbit than North-face LTR thrusters. This table shows that the theoretical constraints facing LTR NSSK maneuvers are indeed seen in reality. The LTR thrusters are far less effective than the AJTs, requiring over three times more fuel to impart three times less acceleration on the spacecraft. The LTR thruster pairs were also unable to fire for as long as the AJT pairs, as the LTR thrusters have a much longer moment arm than the AJTs so minute imperfections in thruster positioning, pointing, or force balance are amplified in comparison with AJT pairs. That amplification leads to higher rates of momentum buildup, meaning the maneuvers need to be shorter lest the spacecraft hit operational momentum limits.

## Conclusion

At the beginning of October, 2021, following a recommendation by Lockheed Martin, GOES-16 transitioned from using LTR thrusters to using "imbalanced pairs" of AJTs to perform NSSK

maneuvers. This type of maneuver uses AJTs which are not diagonally symmetric across the spacecraft's center of mass, typically using AJT13 and AJT16 (see Figure 2). Imbalanced AJT maneuvers rely on the increased efficiency of the AJTs to make up for the momentum imbalance created by using pairs that are not symmetrically positioned about the spacecraft's Y-axis. However, because the AJT13/16 pair is so close to the spacecraft's Y-axis, and because the moment arms are balanced in the spacecraft's X-axis and are very small in the spacecraft Z-axis, that momentum imbalance is an acceptable tradeoff for the increased fuel efficiency of using AJTs over LTR thrusters.

Creating NSSK maneuvers which use imbalanced AJT pairs is far simpler than creating LTR NSSK maneuvers in *FocusSuite*, even though imbalanced AJT pairs are also not natively supported by the *MANCMD* module. However, the techniques for creating LTR NSSK maneuvers live on: the most complex steps for creating imbalanced AJT NSSK maneuvers are accomplished using code copied directly from the LTR NSSK-generation routines. GOES-16 is expected to continue using imbalanced AJTs for its inclination control for the foreseeable future, but LTR NSSK maneuvers remain a contingency option if they are ever deemed necessary.

## APPENDIX: ACRONYMS

| | |
|---|---|
| AJT | ArcJet Thruster |
| COTS | Commercial Off-The-Shelf |
| EWSK | East/West Station-Keeping |
| GOES-R | Geostationary Operational Environmental Satellite – R series |
| GS | Ground System |
| IMF | Impulsive Maneuver File |
| LTR | Low-Thrust REA |
| MA | Momentum Adjust |
| MOST | Mission Operations Support Team |
| NASA | National Aeronautics and Space Administration |
| NOAA | National Oceanic and Atmospheric Administration |
| NSSK | North/South Station-Keeping |
| RWA | Reaction Wheel Assembly |
| WR | Work Request |

## APPENDIX: SAMPLE *SOL* SCRIPT[6]

```
define procedure getarrayelement
   & inputs are Array Position
   & outputs are Element

   # Use an internal array
   set Array_Int to Array

   # Initialize the counter
   set iI to 1

   # Loop through the array
   repeat for each ElementArray in Array_Int

      # If the correct index was found, save the value
      if (iI = Position) then
         Set Element to ElementArray
         Break
      end if

      # Check the next index
      increment iI by 1
   end repeat
end define
```

## REFERENCES

[1] NASA Safety Center, *SMSR Integrated Master Schedule*, June 7, 2021

[2] Chu, D., Chen, S., Early, D., Freesland, D., Krimchansky, A., Naasz, B., Reth, A., Tadikonda, K., Tsui, J., Walsh, T., "GOES-R Stationkeeping and Momentum Management", *29th Annual AAS Guidance and Control Conference*, paper 06-046, Breckenridge, CO, 2006

[3] Chapel, J., Stancliffe, D., Bevacqua, T., Winkler, S., Clapp, B., Rood, T., Freesland, D., Reth, A., Early, D., Walsh, T., Krimchansky, A., "In-Flight Guidance, Navigation, and Control Performance Results for the GOES-16 Spacecraft", *10th International ESA Conference on Guidance, Navigation & Control Systems*, Salzburg, Austria, 2017

[4] Freesland, D., Chu, D., Reth, A., Krimchansky, A., Donnelly, M., Comeyne, G., "Advancing the Next Generation GOES-R Operational Availability", *28th Annual AAS Guidance and Control Conference*, paper 05-006, Breckenridge, CO, 2005

[5] GOES-R Series Concept of Operations, p. 41, June 2020

[6] GMV Aerospace and Defense S.A.U., "Autofocus Software User Manual", p. 60, 2015